**10. Egy táblajáték program tesztelését kell elvégezni. Feladata, hogy a béta állapotú szoftverben hibákat tárjon fel.**

* Tegyen javaslatot egy teljes körű tesztelési tervre!
* Határozza meg a „jó teszteset” kritériumait!
* Tegyen javaslatot a tesztelési dokumentum formájára és felépítésére!
* Ismertesse egy hibakezelő rendszer működését!

A táblajáték program tesztelésének célja a béta állapotú szoftverben lévő hibák feltárása. A tesztelést úgy kell elvégezni, hogy a tesztelő elvégzi a játék minden lehetséges lépését, és ellenőrzi, hogy a program megfelelően reagál-e minden esetben. Ha a program valamilyen hibát mutat, a tesztelőnek rögzítenie kell azt, hogy azt később a fejlesztők javíthassák.

Egy teljes körű tesztelési terv a következő lépéseket tartalmazná:

1. Részletesen meg kell vizsgálni a táblajáték szabályait, hogy a tesztelő ismerje azokat.
2. Készíteni kell egy teszteset-listát, amely tartalmazza a játék összes lehetséges lépését és azok várható eredményeit.
3. Az egyes teszteseteket sorban kell végrehajtani, ellenőrizve, hogy a program megfelelően reagál-e minden esetben. Ha valamilyen hibát észlelnek, a tesztelőnek rögzítenie kell azt.
4. A tesztelés során rögzített hibákat a fejlesztőknek ki kell javítani, és az újra tesztelni kell, hogy biztosak lehessenek benne, hogy a hibák javítva lettek.
5. A tesztelés befejezése után el kell készíteni egy jelentést, amely tartalmazza a tesztelés során talált hibákat és azok javítását.

Egy jó tesztesetnek a következő kritériumoknak kell megfelelnie:

1. Pontosan definiált kezdőállapotot és végső állapotot kell tartalmaznia, amelyeket a tesztelőnek reprodukálnia kell.
2. A tesztesetnek tartalmaznia kell minden olyan lépést, amelyet a program végrehajthat a kezdőállapotból a végső állapotba.
3. A tesztesetnek részletesen meg kell határoznia a várható eredményt minden egyes lépés után, hogy a tesztelő ellenőrizhesse, hogy a program megfelelően reagál-e.
4. A tesztesetnek olyan hibákat kell felmutatnia, amelyeket a tesztelő nem várt.
5. A tesztesetnek olyan részleteket kell tartalmaznia, amelyek lehetővé teszik a fejlesztők számára, hogy a hibát reprodukálják és javítsák.

A tesztelési dokumentumot általában egy táblázat formájában kell elkészíteni, amelyben az egyes teszteseteket sorban feltüntetik, valamint a tesztesetekhez tartozó kezdőállapotokat, lépéseket és várható eredményeket. A tesztelési dokumentum felépítése a következő lenne:

1. Bevezetés: rövid áttekintés a tesztelés céljáról és módszereiről.
2. Teszteset-lista: sorban feltünteti a teszteseteket és azok rövid leírását.
3. Tesztesetek részletes leírása: részletes leírás minden egyes tesztesethez tartozó kezdőállapotról, lépésekről és várható eredményről.
4. Teszteredmények: rögzíti a tesztelés során talált hibákat és azok javítását.
5. Összegzés: rövid összefoglaló a tesztelés eredményeiről.

Egy hibakezelő rendszer a programban felmerülő hibák kezelésére szolgál. A rendszer általában a következő lépéseket követi:

1. A program futása során, ha valamilyen hiba történik, a rendszer megállítja a program futását és hibajelzést küld a felhasználónak.
2. A felhasználó rögzítheti a hibát a rendszerben, megadva annak részletes leírását és a hiba lokalizálásához szükséges információkat.
3. A rendszer továbbítja a hibajelentést a fejlesztőknek, akik megvizsgálják a hibát és javítják azt.
4. A javított programot újra tesztelni kell, hogy biztosak lehessenek benne, hogy a hiba javítva lett.
5. Ha a tesztelés sikeres volt, a javított programot kiadhatják a felhasználók számára. Ha további hibák merülnek fel, a fenti lépések ismét megismétlődnek, amíg a program hibamentes nem lesz.